A Slot-Asynchronous MAC Protocol Design for Blind Rendezvous in Cognitive Radio Networks

Xingya Liu and Jiang Xie
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte
Email: {xliu33, Linda.Xie}@uncc.edu

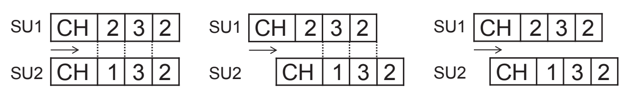
***Abstract*—In cognitive radio networks (CRNs), two users have to rendezvous on a common available channel before communications. Most existing rendezvous papers focus on the channel-hopping (CH) sequence design. However, rendezvous may suffer from the handshake failure on the rendezvous channel, especially in unsynchronized-slot scenarios. In this paper, the challenge of slot-asynchronous rendezvous in CRNs is addressed for the first time. A protocol aiming to improve the handshake performance during the CH process is proposed. By analyzing the potential factors leading to the handshake failure, we design a novel MAC protocol with an optimal size of a time slot which can mitigate the effects of these factors and provide the shortest time for rendezvous. In addition, we also propose a probabilistic model for estimating the average rendezvous time under different CRNs. Simulation results validate our analytical model and demonstrate that our proposed protocol can achieve the rendezvous time close to the theoretical value under slot-asynchronous scenarios.**

I. Introduction

In order to solve the spectrum scarcity and under-utilization problem, cognitive radio emerges as a promising technology which allows a secondary user (SU) to access the spectrum unoccupied by licensed users, or, primary users (PUs). It also requires a SU to vacate channels for the returning of PUs. In other words, the available channels for a SU may change by time or its location. Hence, unlike traditional wireless networks, a control channel that is commonly available to all SUs in a cognitive radio network (CRN) may not exist or cannot last for a long time. It is also impractical for a SU to obtain other's channel information using such a common control channel (CCC). Therefore, two SUs meeting each other on a common channel is a basic step before they can establish communications in CRNs. This process is called blind rendezvous.

To achieve blind rendezvous, the sequence-based channel-hopping (CH) technique can be used. In this approach, each SU first senses the spectrum and generates a set of available channels. Then, it hops onto these channels one by one following a predefined sequence. Thus, two SUs can rendezvous if they hop on a same channel at the same time. The state-of-the-art CH [1] can guarantee the rendezvous between any two SUs if they have at least one common available channel. It mainly focuses on designing the CH sequence to achieve rendezvous in a short time period which is called time to rendezvous (TTR). TTR represents the number of channels a SU needs to go through before hopping to a same channel with another SU. However, in practical scenarios, successful hopping on a same channel does not necessarily lead to a successful handshake which can be affected by many factors [2]. Only after a successful handshake, can two SUs truly establish data communications. Thus, we define time to handshake (TTH) as our performance metric in this paper. In order to get a shorter TTH, existing hopping algorithms need to work under appropriate MAC protocols to guarantee successful handshake in CRNs.

In such a MAC protocol, the key feature to support the CH is to maintain per-unit-length the same in all sequences, i.e., each SU's sojourn duration on each channel should be the same. This feature accords with the time-slotted system where the staying time on each channel can be treated as one time slot. A time slot should be long enough for two SUs to complete a handshake process. It means that a Request-to-Send (RTS) and a Clear-to-Send (CTS) can be successfully exchanged by the sender and the receiver, based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in IEEE 802.11. On the other hand, the time slot of each SU might need to be synchronized in order to ensure that two SUs can hop on the same channel at the same time. Although some CH papers [1], [3] claim that their sequence design can work under the asynchronous scenario, it differs from the asynchronous case in the protocol design. For example, Fig. 1(a) shows a synchronous CH case where SU_1 and SU_2 start to hop at the same time; Fig. 1(b) shows the asynchronous CH case where two SUs start to hop at different time slots; and in Fig. 1(c), the slots are unsynchronized. We name the latter two cases as user-asynchronous case and slot-asynchronous case, respectively. In this paper, we focus on the slot-asynchronous case.



(a) synchronous (b) user-asynchronous (c) slot-asynchronous

Fig. 1. Synchronous case and asynchronous cases in different designs.

However, not all MAC protocols in CRNs can be used for blind rendezvous, such as the protocols under the single channel [4] and the underlay model [5], [6]. Other designs [7], [8] either directly use the synchronized time slot assumption [9]–[12] or achieve synchronization by impractical methods, including employing a CCC [13]–[17], using multiple transceivers [12], [13], [17]–[19], or broadcasting beacons on all channels before rendezvous [20], which is less efficient due to high overhead and collision. The protocol proposed in [21] is claimed to be robust enough under asynchronous slots. However, it does not fully consider the potential problems in such a scenario and lacks details. In summary, all the existing MAC designs consider time synchronization as a necessary condition for blind rendezvous and achieve it in different impractical ways.

In this paper, we consider achieving rendezvous from a different perspective. We propose a novel RTS/CTS handshake mechanism to mitigate the effects caused by asynchronous time slots. This mechanism can achieve successful handshake with a high probability when the sender and the receiver arrive on a same channel at different moments due to their asynchronous time slots. Meanwhile, this mechanism can also solve the problems prohibiting successful handshake in synchronous scenarios. The length of a time slot plays a crucial

This work was supported in part by the US National Science Foundation (NSF) under Grant No. 0953644, 1218751, and 1343355.

role in this design. It is a tradeoff parameter. When a time slot is long, the probability of having a successful handshake is high, but it takes a long time for two SUs to hop on a same channel. On the other hand, if a time slot is short, the handshake process cannot be guaranteed, which may cost more time to get a successful rendezvous. Therefore, we also obtain the optimal length of a time slot for our design in terms of the shortest TTH. Simulation results validate the optimality of the time slot and indicate that the synchronized slot assumption is not necessary for our rendezvous protocol design in CRNs. To the best of our knowledge, this is the first work on MAC design for blind rendezvous in CRNs under the slot-asynchronous scenario.

The rest of this paper is organized as follows. In Section II, we analyze three problems that may happen during one time slot which can potentially ruin a handshake process. At the same time, we design an appropriate protocol to mitigate the effect of these problems as much as possible. In Section III, we derive the expression of the optimal length of a time slot which can minimize the TTH. Simulation results are shown in Section IV, followed by the conclusions in Section V.

II. Problem Analysis and Protocol Design

In this section, we first introduce the system model in our analysis. Then, we analyze three main problems that may result in handshake failure in one time slot and at the same time establish our protocol design to solve these problems.

A. System Model

The system considered in this paper consists of finite number of randomly distributed PUs and SUs which can operate over a set of orthogonal channels. Packet arrivals of both PUs and SUs follow the Poisson distribution. Each PU is randomly assigned a channel when a new packet needs to be transmitted. Each source SU randomly chooses a SU within its transmission range as its destination SU when a new packet needs to be transmitted. Both the source SUs and listening SUs sense all channels to find their own available channels and utilize a CH algorithm to hop among them. This process should be done periodically to avoid channel status change due to PU activities.

We assume that each SU works in a half-duplex mode. When a source SU wants to communicate with its destination SU, the sender sends an RTS message on each channel it hops on during each time slot until receiving the correct CTS. We call the SU in this process an **active SU**. On the other hand, since a passive SU who has no packet to send may become a potential destination SU of its neighbor, it keeps listening on each channel it hops on during each time slot until receiving a correct RTS.

The handshake in our model is assumed to be free of propagation-interference loss. Compared with traditional handshake failures, a more possible reason for a handshake failure in CRNs is that the destination SU is not on the same channel in the current time slot. Hence, for the sake of quick rendezvous, the source SU should leave its current channel and keep hopping on to other channels if the corresponding CTS is not received at the end of a time slot. Only after a successful handshake, can the data transmission take place. The total time a source SU spends before a data packet being transmitted is the **TTH**. Since this paper focuses on the handshake process, we denote the TTH as our **service time** in the queuing system analysis.

Moreover, even the destination SU is on the same channel with the source SU, they may still fail to handshake. The first reason is that an RTS cannot be received completely due to asynchronous time slots. The second reason is neighbor SU's interference, including channel contention and the hidden terminal collision. The last reason is called both-shouting problem caused when the destination SU is also in an active mode. *The latter two issues also exist under synchronous scenarios*.

B. Analysis of the Failure Receiving Problem

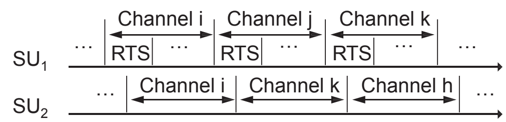


Fig. 2. The RTS failure receiving cases.

In an asynchronous CRN, as illustrated in Fig. 2, a passive SU may not receive a complete RTS due to hopping onto a potential rendezvous channel later than the starting time of an RTS sending (on channel i), or leaving earlier before the sending finishes (on channel k).

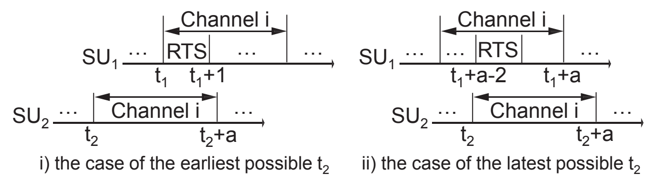
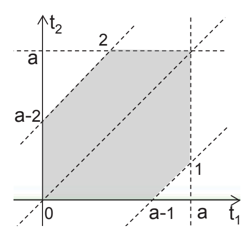


Fig. 3. The cases that at least one RTS can be completely received.

Let SU_1 and SU_2 arrive on a same channel at moments t_1 and t_2, respectively. We normalize the length of sending an RTS/CTS to 1. If the length of a time slot is a, a should be longer than 2 so that at least one pair of RTS and CTS exchange can be completed in a time slot. In addition, we have the constraint $|t_1 - t_2| \leq a$ to ensure that SU_1 and SU_2 have overlapping time on the common channel. There are also the following constraints (see Fig. 3) to help SU_2 hear at least one complete RTS from SU_1. If $t_1 \geq t_2$ (SU_1 hops on the channel later than SU_2), the leaving time of SU_2 should be at least later than the end time of the first RTS sent by SU_1 on the common channel, i.e., $t_2 + a \geq t_1 + 1$. If $t_1 \leq t_2$ (SU_1 hops on the channel earlier than SU_2), the arriving time of SU_2 should be at least earlier than the start time of the last possible RTS sent by SU_1 on the common channel, i.e., $t_2 \leq t_1 + a - 2$. Note that after each RTS is sent out, a SU must wait for a while for the potential CTS. Thus, the last RTS in the current time slot must be sent before $a - 2$.

To sum up, we have the following equivalent inequalities and their corresponding graphic illustration:

$$\begin{cases} 0 \leq t_1 \leq a \\ 0 \leq t_2 \leq a \\ t_1 - a + 1 \leq t_2 \leq t_1 \\ t_1 \leq t_2 \leq t_1 + a - 2 \end{cases}$$



The above shadow area represents the feasible ranges of t_1 and t_2 to ensure the receiving of at least one complete RTS. Therefore, we can derive the probability that a passive SU receives a complete RTS from another SU on a channel in the asynchronous scenario, P_1,

$$P_1 = \frac{\text{size of the shadow area}}{\text{size of the } a \times a \text{ square}} = \frac{a^2 - 2.5}{a^2}, \qquad a \geq 2. \quad (1)$$

In synchronous CRNs, it is natural to define the size of one time slot to be the length of an RTS and a CTS exchange for the sake of quick rendezvous, i.e., $a = 2$. However, according to (1), this design leads to a probability of 0.375 to have a successful RTS reception even after two SUs hop on a same channel in the asynchronous scenario. Since P_1 is a monotonically increasing function of a, from this point of view, we should design the length of a time slot as long as possible and let a SU keep sending RTS until the current time slot ends.

C. Analysis of the Neighboring Interference Problem

In CRNs, especially in cognitive radio ad hoc networks (CRAHNs), several other SUs may be within a SU's sensing range. Hence, three or more SUs may hop on a same channel in one time slot during their rendezvous processes. They may interfere with each other in two scenarios. One is the presence of RTS collisions in the hidden terminal case. The other is the continuous contention for sending RTS between active neighboring SUs in one time slot.

In traditional wireless networks, one reason that an active node cannot receive the correct CTS is the RTS collision from a hidden terminal. Thus, 802.11 CSMA/CA requests a node to perform a binary exponential backoff when experiencing the absence of CTS. However, this mechanism may not increase the successful rate of handshake when applied to CRNs, since a more possible reason for the absence of CTS is that the destination node is not on the same channel. In addition, to support backoff, the size of a time slot needs to be unacceptable long. Moreover, the backoff SU may still collide with a new SU who just hops on this channel after the backoff under the asynchronous scenario. On the other hand, each time when a SU resends an RTS, it is an additional contender for other SUs. If the destination SU is not on the same channel, the source SU keeps rejoining the contention, which affects other SU's opportunities to send the RTS.

For example, in Fig. 4, SU_1 has successfully sent an RTS several times in a time slot (gray RTS/CTS means the supposed sending/receiving but not achieved). If SU_2 is a hidden terminal of SU_1, it will collide with SU_1's jth resend. If SU_2 is a neighbor of SU_1, it will lose the opportunity to send its RTS because of SU_1's kth resend. If SU_1's destination SU is absent on this channel during this time slot, this contention keeps happening till SU_1 leaves the channel.

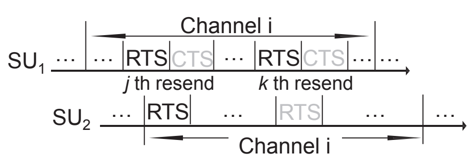


Fig. 4. The neighboring interference cases.

Therefore, the traditional method for resolving the RTS collision is not desirable for asynchronous rendezvous in CRNs. A better mechanism is required in our protocol. From Section II-B, note that P_1 is only affected by three factors: the sending moments of the first and the last RTS in a time slot, and the length of a time slot. Based on this observation, we redesign the protocol which can solve the neighboring interference problem and meanwhile has an equivalent effect as the design in Section II-B.

We propose that an active SU only sends an RTS twice in a time slot: one at the beginning and one at the end of a time slot if a channel is idle, and listens to the channel during other periods, as the SU_1 illustrated in Fig. 5. However, if the length of a time slot, a, is not long enough for sending the second RTS (with CTS receiving), a SU gives up the resend and listens to the channel until the current time slot ends. Thus, we design the length of a time slot to be either $a > 4$ (neglect the lengths of DIFS, SIFS, and the contention-window in 802.11) or $a = 2$ (without resending and contention mechanisms in a time slot). On the other hand, if a SU senses a channel busy, it waits to send the first RTS until the channel idle long enough for a CTS time, as the SU_2 in Fig. 5. The gap between the dashed line and the solid line is the time for DIFS and the backoff time for contention. We neglect these obligatory frames in our analysis.

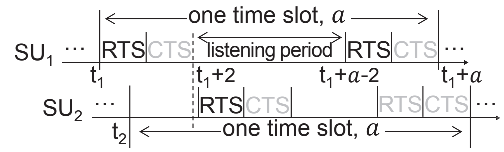


Fig. 5. The revised resending mechanism.

Let SU_2 be a source SU and SU_1 be its neighbor. Let t_2 and t_1 be their arrival times on a same channel. Assume that their destination SUs are not on the same channel. If $t_2 < t_1$, SU_2 can send its RTS in this time slot. If $t_2 \geq t_1$, in the $a = 2$ case, SU_2 can send its RTS if arriving after SU_1 finishing its RTS sending, i.e., $t_2 \geq t_1 + 1$. Using the same constraint $|t_1 - t_2| \leq a$ and solving the inequalities the same way as P_1, the probability that a SU can successfully send an RTS with a neighbor SU on the same channel is 0.625 when $a = 2$. In the $a > 4$ case, even SU_2 arrives during SU_1's first RTS sending time as in Fig. 5, SU_2 can still send its RTS as long as the moment it starts to send is earlier than SU_1's second RTS sending, i.e., $t_1 + 2 < t_1 + a - 2$, which always stands since $a > 4$. Moreover, if SU_2 arrives during SU_1's second RTS sending ($t_1 + a - 2 < t_2 < t_1 + a - 1$), we have $t_2 > t_1 + 2$ when $a > 4$, or, $t_2 + a - (t_1 + a) > 2$. In other words, SU_2 has enough time for sending its RTS after SU_1 leaves the channel. Hence, the probability that a SU can successfully send an RTS, P_2, increases to 100% when $a > 4$. Therefore,

$$P_2 = \begin{cases} 0.625, & a = 2 \\ 1, & a > 4 \end{cases}. \quad (2)$$

This design also reduces the RTS collision rate due to the low RTS sending frequency. Compared with the traditional design, the listening period in the middle of a time slot provides the opportunity for another SU to send an RTS without collision. Suppose that SU_2 is a hidden terminal of SU_1. Then, the case that SU_1 can successfully send at least one RTS without collision is when the RTS from SU_1 has no overlap with the RTS from SU_2, i.e., $t_1 + 1 < t_2$, or $t_1 + a - 2 > t_2 + a - 1$. We name this probability P_3. We derive it in a similar way as P_1,

$$P_3 = \frac{(a-1)^2}{a^2}, \qquad a = 2 \text{ or } a > 4. \quad (3)$$

It also requires the a to be as long as possible to achieve the collision-free sending.

D. Analysis of the Both-Shouting Problem

In Section II-B, we consider the failure receiving problem when the destination SU is a passive SU. However, it may also be another active SU. Consequently, when a SU is hopping and searching for its destination SU, the target SU may also be searching for another SU. An extreme case is that they are searching for each other, which is a deadlock when $a = 2$.

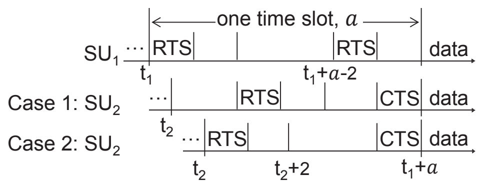


Fig. 6. Two successful cases when $t_1 < t_2$ under the both-shouting scenario.

Let SU_1 be a source SU and SU_2 be the target SU which is also an active SU when it arrives on the same channel. If $t_1 < t_2$, there are two cases in which SU_2 can hear a complete RTS from SU_1. Case 1 is that SU_2 arrives during SU_1's sending its first RTS. As illustrated in Fig. 6, based on our designed protocol from Section II-C, after waiting for a CTS-long idle period, SU_2 starts to send its own RTS. If SU_2's target SU is SU_1, SU_1 replies a CTS and they begin to transmit data. In other words, the deadlock case can be easily solved to have a successful handshake when $a > 4$. If SU_2's target SU is not SU_1, then SU_1 waits until the moment t_1+a-2. If the channel is idle, SU_1 sends its last RTS and SU_2 will hear the last RTS from SU_1. In this case, it requires that $t_1 + 4 \leq t_1 + a - 2$ (enough time to send the last RTS), or, $a \geq 6$. Case 2 is that SU_2 arrives after SU_1's first RTS ($t_2 \geq t_1+1$). SU_2 senses the channel idle and sends its first RTS after arriving. Similarly, SU_1 overhears this RTS and waits until $t_2 + 2$ if it still has time to send its last RTS, i.e., $t_2 + 2 \leq t_1 + a - 2$. It requires that $a \geq 5$ in this case.

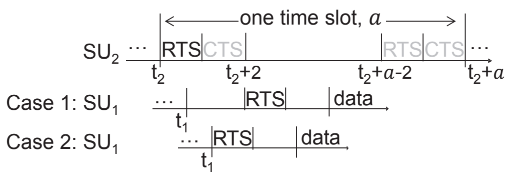
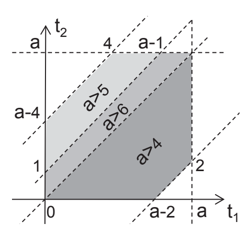


Fig. 7. Two successful cases when $t_1 > t_2$ under the both-shouting scenario.

If $t_1 > t_2$, we also analyze two cases, as illustrated in Fig. 7. In case 1, $t_2 < t_1 < t_2+1$, SU_1 starts to send at t_2+2, which should before SU_2's second RTS sending ($t_2+2 \leq t_2+a-2$). Then, $a > 4$ is required for this case to ensure a successful RTS reception at SU_2. In case 2, $t_2 + 1 \leq t_1$, SU_1 can send its first RTS after arriving. SU_2 will hear this RTS as long as it is sent before SU_2's last RTS ($t_1 \leq t_2 + a - 2$). This case requires that $a \geq 3$. In both cases, SUs end their current time slot immediately once the handshake is finished. To sum up, we have the following equivalent inequalities and their corresponding graphic illustration:

$$\begin{cases} 0 \leq t_1 \leq a \\ 0 \leq t_2 \leq a \\ t_1 \leq t_2 \leq t_1 + 1 \\ t_1 + 1 \leq t_2 \leq t_1 + a - 4 \\ t_2 \leq t_1 \leq t_2 + a - 2 \end{cases}$$



The above shadow area represents the feasible ranges of t_1 and t_2 under different values of a. Therefore, we can derive the probability that a SU's RTS can be heard by its destination active SU on the same channel, P_4, which is also a monotonically increasing function of a.

$$P_4 = \begin{cases} 0, & a = 2 \\ (a^2 - 4)/2a^2, & 4 < a < 5 \\ (2a^2 - 2a - 19)/2a^2, & 5 \leq a < 6 \\ (a^2 - 10)/a^2, & a \geq 6 \end{cases}. \tag{4}$$

III. Proposed MAC Protocol

We elaborate the complete protocol in details and derive an optimal a in terms of the minimum TTH in this section.

A. Protocol Details

The overall flow chart for our proposed MAC protocol is presented in Fig. 8. In the initial period, a SU senses all channels and collects its available channel set. Existing sequence/probabilistic-based CH designs can be employed in this step to generate the CH sequence. Then, the SU tunes its radio to the ordered channel and begins a new time slot. During a time slot, any SU can become a destination node once it receives an RTS carrying its ID as the receiver. If the SU also has data to send, it postpones its own data in queue and receives other's transmission first. On the other hand, a passive SU can become a source node once it has data to send. In synchronous environments, it has to wait till the next time slot to change its role. However, in our design, it sends an RTS immediately if there is still time left in the current time slot since there is no need for slot-synchronization. Once a pair of SUs completes the handshake in a time slot, they stay on the same channel transmitting data until they finish the communication. When the pair detects a PU presence on the channel, a spectrum handoff [22] is performed for resuming the transmission on another channel.

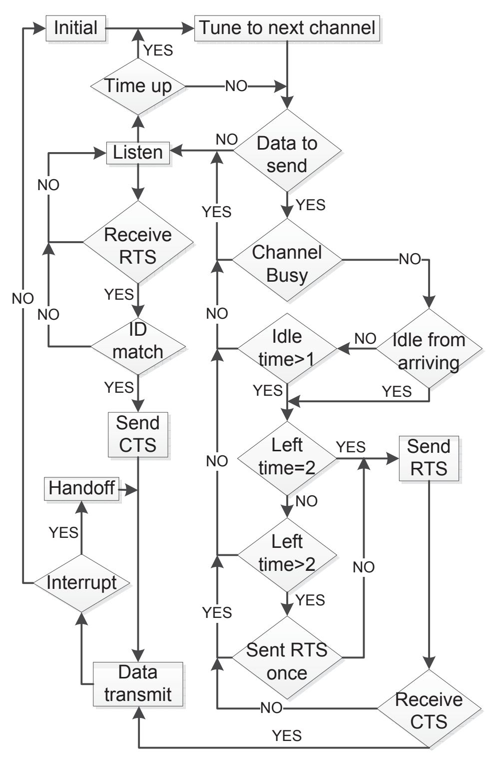


Fig. 8. The flow chart of the proposed MAC protocol.

In the figure, “Left time = 2” refers to the moment approaching $a-2$. The backoff for the last RTS sending is counted in a reverse time from $a-2$. For example, if a random number 0.2 is generated for the backoff time, the last RTS will be sent from the moment $a-2-0.2$. The use of SIFS and DIFS in our protocol is the same as in 802.11 MAC.

B. Optimal Time Slot

When considering the whole rendezvous process, many time slots are needed before a pair of SUs hop on a same channel and have a successful handshake. We denote P as the probability that a source SU successfully handshakes with its destination SU in the next time slot. Use $\overline{X}$ to represent the average service time (TTH) of a SU. Then,

$$\overline{X} = a\sum_{i=1}^{\infty} i(1-P)^{i-1}P = \frac{a}{P}. \tag{5}$$

From the analysis in Section II, a long time slot can improve the successful handshake rate in one time slot. Thus, a and P are positive correlated. Therefore, an optimal a is needed for (5) in terms of the shortest $\overline{X}$. We derive the optimal a as follows.

First, let ρ represent the **active rate** of a SU, i.e., the probability that a SU is in an active mode. Assume that the data traffic is homogeneous in the secondary network, i.e., the active rate of a SU is the same everywhere in the network. Let λ be the average packet arrival rate of a SU (in the unit of one RTS sending time). Then, $\rho = \lambda\overline{X}$.

We further denote P_0 as the probability that a source SU successfully hops on a same channel with its destination SU in a time slot. P_0 varies under different CH designs [23]. If we do not consider the neighboring interference problem, we can derive P as P_I (P in an idle environment):

$$P_I = P_0(1-\rho)P_1 + P_0\rho P_4, \tag{6}$$

where P_1 and P_4 are the same probabilities defined in Section II-B and II-D when the destination SU is passive or active, respectively.

However, the neighboring-inference problem cannot be ignored when a SU is in a dense network where the number of its neighbors is large. Assume that there are an average of K neighbors of a SU. Excluding the destination SU, the number of the potential contenders of a source SU is $K_1 = K-1$. We denote K_2 as the average number of its hidden terminal SUs. Then,

$$\begin{aligned} P =& Pr(K_1 = 0,\ K_2 = 0)P_I \\ &+ Pr(K_1 = 1,\ K_2 = 0)P_2P_I \\ &+ Pr(K_1 = 0,\ K_2 = 1)P_3P_I + ... \end{aligned}$$

where $Pr(K_1 = 0,\ K_2 = 0)$ is the probability that no neighbor and hidden terminal exists on the same channel during the source SU's one time slot, i.e., $(1-P_0\rho)^{K_1+K_2}$. We can further derive other probabilities regarding different values of K_1 and K_2. In this way, P can be written as:

$$\begin{aligned} P \approx& (1-P_0\rho)^{K_1+K_2}P_I \\ &+ \binom{K_1}{1}P_0\rho(1-P_0\rho)^{K_1+K_2-1}P_2P_I \\ &+ \binom{K_2}{1}P_0\rho(1-P_0\rho)^{K_1+K_2-1}P_3P_I. \end{aligned} \tag{7}$$

We do not consider the cases when $K_1+K_2 > 1$ due to two reasons. One is that the probabilities of $K_1+K_2 > 1$ are negligible due to the $P_0\rho$ part. P_0 is usually on the order of $\frac{1}{M}$, where M is the total number of channels in a CRN [1], [3], [23]. Meanwhile, ρ should be small enough in CRNs to avoid network congestion [2]. Then, $P_0\rho$ is a quite small value. Moreover, the probability that $K_1+K_2 \geq 2$ involving $(P_0\rho)^2$ or higher power can be neglected. The other reason is that the probabilities of successful handshake under $K_1+K_2 > 1$ are also negligible, referring the derivation part of P_2 and P_3 in Section II-C.

From (5)-(7), we can get (8)

$$a \approx \overline{X}P_I(1-P_0\lambda\overline{X})^{K_1+K_2-1}\left[1-P_0\lambda\overline{X}(1-K_1P_2-K_2P_3)\right]. \tag{8}$$

It is a transcendental equation because of independent variables K_1 and K_2. Once the network parameters K_1, K_2, M, and λ are given, the expression of $\overline{X}$ in terms of a can be derived and the optimal a that minimizes $\overline{X}$ can be obtained. Detailed examples are given in Section IV.

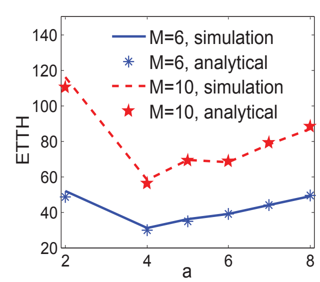
IV. Performance Evaluation

In this section, we evaluate the performance of our proposed MAC protocol under different scenarios by comparing simulation results with the analytical values. In our simulation, we assume that the packet arrival of each SU follows the Poisson distribution. Moreover, since P_0 is a variable independent of our analysis, we adopt the random CH algorithm under which P_0 is exactly $\frac{1}{M}$ in order to easily adjust the value of P_0. Additionally, we consider a grid network where $K_1 = 3$ and $K_2 = 3$. More importantly, each SU in the simulation has its own clock and is not required to be synchronized with others. Other parameters used in our simulation are listed in Table I.

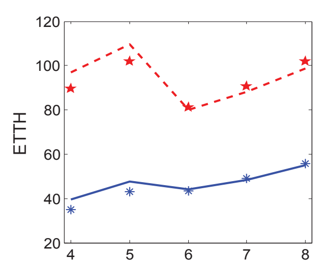
TABLE I. Simulation Parameters

Number of SUs	64 (8×8 grid)
Channel data rate	2 Mbps
The size of (RTS+CTS)	$160 + 112$ bits (802.11b/g)
Simulation time	10000

Fig. 9 illustrates the expected TTH (ETTH) of the whole network under different numbers of channels. The simulation results match the analytical results very well with a maximum difference of 5%. Fig. 9(a) shows that $a = 4$ is the optimal size of a time slot when the average packet arrival rate is low, or, the active rate of a SU is low ($\lambda = 50$ pkt/s, $\rho \approx 0.1 - 0.2$). Since ρ is small, there are more idle time slots during rendezvous. Consequently, the advantages of a large a ($a > 4$) when dealing with complicated cases (P_2, P_3, and P_4) cannot be fully utilized. Therefore, the increasing rate of the ETTH after $a = 4$ is higher when there are more channels to hop ($M = 10$).



(a) $\lambda = 50$ pkt/s.



(b) $\lambda = 100$ pkt/s.

Fig. 9. ETTH vs. a in different traffic conditions

Fig. 9(b) shows the impact of different a in a nearly saturated network. When M is small, $a = 4$ still holds the optimal size of a time slot. However, note that the ETTH when $a = 6$ is already a bit lower than when $a = 5$. This means that the advantages of a large a become dominant in the results. In the $M = 10$ case, the optimal size is when $a = 6$ (even when $a = 7$ has the same effect as when $a = 4$). Furthermore, the design of $a = 2$ cannot stand under this scenario. This is because when $a = 2$, the low probability of the handshake successful rate increases the TTH. Then, the long TTH leads to a high ρ which further results in $P_4 = 0$ and an infinite TTH. On the other hand, from (1)-(4), the improvement of each probability becomes less and less when a is larger than 6. It is also shown in Fig. 9 and 10 that the ETTH monotonically increases after $a = 6$.

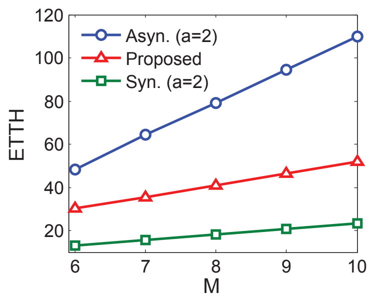


Fig. 10. Compare with the MAC without our design in different scenarios.

Fig. 10 compares the performance of different MAC protocols under the same traffic condition ($\lambda = 50$ pkt/s). Since we already derive the optimal size of a time slot under such scenario, the proposed line is the performance equipped with our MAC with $a = 4$ over different M. The asynchronous line belongs to the random CH protocol with the traditional MAC under asynchronous scenarios. The performance of this traditional MAC under the synchronous slot scenario is shown as the square-line. From Fig. 10 we can see that the proposed MAC performs much better than traditional MAC and closer to the synchronous one (the ideal case).

TABLE II. TTH VS. TTR

M (λ)	6 (50)	6 (100)	10 (50)	10 (100)
ETTH (in unit of slots)	7.75	8.95	14.32	13.50
ETTR (theoretical)	6	6	10	10

To obtain the TTH in the unit of slots, we divide the minimum TTH using its corresponding a. For example, the minimum TTH for the case where $M = 10$ and $\lambda = 100$, is $80.99/6 = 13.50$ slots. Then, the average numbers of time slots a SU spent under different scenarios are shown in Table II. It is shown that our proposed MAC protocol under slot-asynchronous scenarios can maintain the ETTH with the theoretical ETTR. Therefore, using our proposed protocol, it is not necessary to have slots synchronized in CRNs.

V. Conclusion

In this paper, we developed probabilistic models for each possible factor which may influence the successful handshake during blind rendezvous. Then, according to the analysis of each factor, we proposed corresponding schemes and integrated them into a novel MAC protocol with the optimal size of a time slot in terms of the shortest TTH. Simulation results verify the optimality of the time slot and show that our design can maintain the rendezvous performance at the theoretical level in different practical scenarios.

References

[1] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE INFOCOM*, 2011.

[2] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," in *Proc. IEEE INFOCOM*, 2014.

[3] Z. Gu, Q.-S. Hua, Y. Wang, and F. Lau, "Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks," in *Proc. IEEE SECON*, 2013, pp. 371–379.

[4] Q. Chen, Y.-C. Liang, M. Motani, and W.-C. Wong, "A two-level MAC protocol strategy for opportunistic spectrum access in cognitive radio networks," *IEEE Transcations on Vehicular Techology*, vol. 60, pp. 2164–2180, 2011.

[5] S.-Y. Lien, C.-C. Tseng, and K.-C. Chen, "Carrier sensing based multiple access protocols for cognitive radio networks," in *Proc. IEEE ICC*, 2008, pp. 3208–3214.

[6] X. Zhang and H. Su, "Opportunistic spectrum sharing schemes for CDMA-based uplink MAC in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 716–730, 2011.

[7] J. Xiang, Y. Zhang, and T. Skeie, "Medium access control protocols in cognitive radio networks," *Wireless Communication and Mobile Computing*, vol. 10, no. 1, pp. 31–49, 2010.

[8] A. De Domenico, E. Strinati, and M. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys and Tutorials*, vol. 14, pp. 21–44, 2012.

[9] Y. Zhao, M. Song, and C. Xin, "FMAC: A fair MAC protocol for coexisting cognitive radio networks," in *Proc. IEEE INFOCOM*, 2013.

[10] D. Hu and S. Mao, "Design and analysis of a sensing error-aware MAC protocol for cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2009.

[11] M. Felegyhazi, M. Cagalj, and J.-P. Hubaux, "Efficient MAC in cognitive radio systems: A game-theoretic approach," *IEEE Transcations on Wireless Communications*, vol. 8, pp. 1984–1995, 2009.

[12] G. Joshi and S. W. Kim, "An enhanced synchronized MAC protocol for cognitive radio networks," in *Proc. IEEE WiCOM*, 2011.

[13] M. Shah, S. Zhang, and C. Maple, "An analysis on decentralized adaptive MAC protocols for cognitive radio networks," in *Proc. IEEE ICAC*, 2012.

[14] L. T. Tan and L. B. Le, "Distributed MAC protocol for cognitive radio networks: Design, analysis, and optimization," *IEEE Transcations on Vehicular Techology*, vol. 60, pp. 3990–4003, 2011.

[15] M. Kalil, A. Puschmann, and A. Mitschele-Thiel, "SWITCH: A multichannel MAC protocol for cognitive radio ad hoc networks," in *Proc. IEEE Vehicular Technology Conference (VTC)*, 2012.

[16] H. Hussein, H. Elsayed, and S. Elramly, "Performance evaluation of cognitive radio network predictive MAC (P-MAC) access algorithm and its enhancement," in *Proc. IEEE ICOIN*, 2013.

[17] S. Jha, U. Phuyal, M. Rashid, and V. Bhargava, "Design of OMC-MAC: An opportunistic multi-channel MAC with QoS provisioning for distributed cognitive radio networks," *IEEE Transcations on Wireless Communications*, vol. 10, pp. 1984–1995, 2011.

[18] C.-F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," in *Proc. IEEE ICC*, 2010.

[19] L. Jiao and F. Li, "A single radio based channel datarate-aware parallel rendezvous MAC protocol for cognitive radio networks," in *Proc. IEEE LCN*, 2009.

[20] S. K. Timalsina, S. Moh, Ilyong Chung, and M. Kang, "A concurrent access MAC protocol for cognitive radio ad hoc networks without common control channel," *EURASIP Journal on Advances in Signal Processing*, vol. 69, 2013.

[21] M. Altamimi, K. Naik, and X. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2010.

[22] Y. Song and J. Xie, "ProSpect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel," *IEEE Trans. Mobile Computing*, vol. 11, no. 7, pp. 1127–1139, 2012.

[23] Y. Song, J. Xie, and X. Wang, "A novel unified analytical model for broadcast protocols in multi-hop cognitive radio ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 13, no. 8, pp. 1653–1667, August 2014.